Analytical Documentation

Release 0.0.1

Read the Docs, Inc.

Nov 14, 2019

Providers

Analytical is a Python library for sending pageviews and events to analytics platforms like Google Analytics except from Python rather than JavaScript so it can be done server side. This has a number of advantages such as working regardless of whether clients block analytics scripts, privacy sensitive information can be anonymized or removed before sending, and it allows sending data only known by the server.

CHAPTER 1

Feature support

- Convenient utilities for anonymizing sensitive information like IP addresses
- Pluggable provider backends for different analytics platforms (currently just Google)

Supports Python 2.7, Python 3.5+, and PyPy.

CHAPTER 2

Example

```
import analytical

provider = analytical.Provider('googleanalytics', 'UA-XXXXXXX-1')
provider.pageview({
    'dl': 'https://example.com',
    'dt': 'My Page Title',
    'ua': 'user-agent',             # User agent
    'uip: '12.34.56.78',            # User IP address
})
```

2.1 Google Analytics

The Google Analytics provider sends data to Google using the Measurement Protocol.

2.1.1 Tracking pageviews

```
import analytical
from analytical.providers.googleanalytics import generate_client_id

ga = analytical.Provider('googleanalytics', 'UA-XXXXXX-Y')
ga.pageview({
    "ua": "user-agent",             # User agent
    "uip": "12.34.56.78",           # User IP address
    "dl": "https://example.com",    # URL of the pageview (required)
    "dt": "page title",             # Title of the page
})
```

For the complete list of parameters, see the reference.

2.1.2 Tracking events

```
import analytical

ga = analytical.Provider('googleanalytics', 'UA-XXXXXX-Y')
ga.event({
    "ua": "user-agent",
    "uip": "12.34.56.78",
    "ec": "event-category",          # Event category (required)
    "ea": "event-action",            # Event action (required)
    "el": "event-label",             # Event label (optional)
    "ev": 0,                         # Event value (optional, integer)
})
```

See the event parameter reference for more details.

2.1.3 Utility functions

analytical.providers.googleanalytics.**generate_client_id**(*user_secret=None*)

Generate a Google Analytics client ID (the `cid` parameter)

GA treats users with the same client ID as the same user for analytics purposes. This function helps generate a client ID that can be used to track new vs. returning visitors without cookies.

```
# Use the User Agent and IP Address
# The downside to this is if the IP or UA changes, it's considered a new user
# The upside is it doesn't require anything from a database, cookies or elsewhere
secret = '{}${}${}'.format('my-secret', ip_address, user_agent)
client_id = generate_client_id(secret)

# Use a user ID value from a database or elsewhere
secret = '{}${}'.format('my-secret', user.id)
client_id = generate_client_id(secret)
```

Parameters **user_secret** (*str*) – a secret that shouldn't change for a given user. If `None`, treat all pageviews and events as new/unique.

Returns **str** a client ID suitable for using with Google Analytics

2.2 Utilities

analytical.utils.**anonymize_ip_address**(*ip_address*)

Anonymizes an IP address by zeroing the last 2 bytes

Zeroing the last two bytes is sufficient to make a user reasonably anonymous while still allowing decent geolocation accuracy. One byte is insufficient for the DoNotTrack standard.

```
anonymize_ip_address('12.34.56.78')  # '12.34.0.0'
```

Parameters **ip_address** (*str*) – the IP address of the user

Returns **str** an anonymized IP address

analytical.utils.**anonymize_user_agent**(*user_agent*)
Anonymizes rare user agents

Currently, this function is pretty naive. Only if the browser or OS family are not recognized, it is considered "rare".

Parameters **user_agent** (*str*) – the browser user agent for the user

Returns str the same user agent or the string "Rare user agent" if user_agent is "rare"

2.3 Frequently Asked Questions

Why track events on the server rather than with JavaScript? One big advantage is that privacy sensitive information such as IP addresses can be anonymized before they are sent to Google or other providers. This also allows tracking events that are only known to the server such as sales data or server side errors.

If I anonymize IP addresses, won't I lose precision in location data? Anonymizing IP addresses will definitely lose precision for IP geolocation. However, by zeroing the last 2 bytes as this library does, IP geolocation should still work to a major city or metro area.

How is this different from other packages like pyga, django-analytical, etc.? Libraries like django-analytical ease the creation of JavaScript tracking tags and don't do server side analytics like this library. Pyga uses the old gajs Google Analytics tracking rather than the more modern, documented, and supported measurement protocol.

2.4 Developing

```
pip install -r development-requirements.txt
pre-commit install
```

2.4.1 Making a release

- Increase the version in analytical/__init__.py.
- Update the changelog in docs/changelog.rst.
- Commit the changes.
- Tag the release in git: git tag $NEW_VERSION.
- Push the tag: git push --tags origin
- Upload the changes to PyPI:

```
rm -rf dist/
python setup.py sdist bdist_wheel
twine upload --sign --identity security@readthedocs.org dist/*
```

2.5 Changelog

v0.0.1 - November 6 2018

- First release

Index

A

G